Statistics of Big Data, Fall 2021-22

Class notes 8

Sources for today's material:
survey by Goldberg et al. on statistical modeling of network data (that appeared in 2010 in the Foundations and Trends in Machine Learning)
An explanation of the pseudo-likelihood approach by Staruss and Ikeda

Network data modeling (ctd.)

The p_1 model of node properties and edge creation

Consider two nodes i, j and the four possible settings of the edges $Y_{ij} \in \{0,1\}$, $Y_{ji} \in \{0,1\}$, as a function of the parameters of the network and the nodes:

- θ: overall rate of connections (like in Erdos-Renyi)
- α_i : *Expansiveness*, measuring how friendly node i is
- β_i : *Popularity*, measuring how attractive node i is
- ρ : *Reciprocity*, measuring how likely $Y_{ij} = Y_{ji}$ is

We also have λ_{ij} a normalization factor. In this setting we can write the four probabilities as a function of the parameters:

$$\begin{aligned}
\log(\mathbb{P}_{ij}(0,0)) &= \lambda_{ij} \\
\log(\mathbb{P}_{ij}(1,0)) &= \lambda_{ij} + \alpha_i + \beta_j + \theta \\
\log(\mathbb{P}_{ij}(0,1)) &= \lambda_{ij} + \alpha_j + \beta_i + \theta \\
\log(\mathbb{P}_{ij}(1,1)) &= \lambda_{ij} + \alpha_i + \alpha_j + \beta_i + \beta_j + \rho + 2\theta
\end{aligned}$$

where λ_{ij} is such that the probabilities sum to 1.

Now note that if we choose $\rho = 0$, $\alpha_i = \beta_i = 0$, $\forall i$, then we get the Erdos-Renyi model with θ only (fixed probability).

To fit this model to data we would write the likelihood as a function of the parameters:

$$\mathcal{L}(\theta, \alpha, \beta, \rho) = C(\lambda) + \sum_{i,j=1, i \neq j}^{N} y_{ij}(\theta + \alpha_i + \beta_j) + \sum_{i<j} y_{ij} * y_{ji} * rho,$$

if we ignore $C(\lambda) = \sum_{i,j} \lambda_{ij}$, then this is an exponential family log-likeilhood and we can find the MLE $\hat{\theta}, \hat{\alpha}_i, \hat{\beta}_i, \hat{\rho}$ with standard approaches. However this is not really accurate — the λ_{ij} are also unknown. However, they are not free parameters, rather complicated functions of the other parameters that violate the exponential family assumption:

$$\lambda_{ij} = -log\left(1 + \exp(\alpha_i + \beta_j + \theta) + \exp(\alpha_j + \beta_i + \theta) + \exp(\alpha_i + \alpha_j + \beta_i + \beta_j + \rho + 2\theta)\right).$$

Note also that α, β, θ are not identifiable in this setting, since we can take either all α, all β or θ and add and subtract constants that sum to zero with no change in the model.

To obtain a proper maximum likelihood solution we can call on more complex optimization approaches, specifically Markov Chain Monte Carlo (MCMC) that we may not have time to discuss in this course that seek a good combination of the parameters for the full likelihood. A more mainstream statistical approach is to use pseudo-likelihood. In this important family of approaches, we define a function that we can optimize and is “similar” to the likelihood but simplified. The main idea here is that if I am given Y_{ji} then the likelihood of Y_{ij} is simple and has a logistic form that does not depend on the $\lambda' s$ which cancel out:

$$\begin{aligned}\mathbb{P}(Y_{ij} = 1|Y_{ji}) &= \frac{\mathbb{P}(Y_{ij} = 1, Y_{ji})}{\mathbb{P}(Y_{ij} = 1, Y_{ji}) + \mathbb{P}(Y_{ij} = 0, Y_{ji})} = \\ & \frac{\exp(\lambda_{ij} + \theta + \alpha_i + \beta_j + Y_{ji}(\theta + \alpha_j + \beta_i + \rho))}{\exp(\lambda_{ij} + \theta + \alpha_i + \beta_j + Y_{ji}(\theta + \alpha_j + \beta_i + \rho)) + \exp(\lambda_{ij} + Y_{ji}(\theta + \alpha_j + \beta_i))} = \\ & \frac{\exp(\theta + \alpha_i + \beta_j Y_{ji}\rho)}{\exp(\theta + \alpha_i + \beta_j + Y_{ji}\rho) + 1},\end{aligned}$$

a regular logistic regression:

$$\operatorname{logit}\left(\mathbb{P}(Y_{ij} = 1|Y_{ji})\right) = \theta + \alpha_i + \beta_j + Y_{ji}\rho,$$

with the linear constraints $\sum_i \alpha_i = \sum_j \beta_j = 0$ for identifiability, which are not a problem (also appear in regular logistic regression with intercept).

So now we have a standard logistic regression model as our maximum pseudo-likelihood solution, and we can also apply the regular logistic regression inference: significance on the parameters, F-tests for model selection, using of AIC and model selection criteria, etc. However, we should keep in mind that there are major problems here:

- We are not doing maximum likelihood, but maximum pseudo-likelihood, so by definition there is no guarantee that the theory on which ML inference is based is relevant. The help for the pstar function we are using even includes a warning:

  > Estimation of p^* models by maximum pseudo-likelihood is now known to be a dangerous practice. Use at your own risk.

- Even if we accept the PML approximation, note that we have $2N + 2$ parameters and N^2 observations (edges) in a naive view, but typically the number of actual edges is more likely $O(N)$, in which case the ML asymptotics, which are for number of parameters fixed, number of observations diverging, is not relevant anyway.

The more general ERG-p^* approach

So far the only complication we assumed is mutuality for directed graphs. Frank & Strauss considered a larger family of graphs, where the probability of an edge can depend on all edges that share a node with it (rather than only the opposite edge between the two nodes). They showed that for undirected graphs all these models can be written in general form:

$$\mathbb{P}(Y = y) = \exp\left(T(y)\tau + \sum_{k=1}^{N-1} S_k(y)\theta_k + \psi(\theta, \tau) \right.$$

This is a model with N parameters where the summary statistics are:

- $S_k(y),\ k = 1, \ldots, N-1$ — the number of k-stars in the graph, where a k-star is a node connected to k neighbors. 1-star is an arc, 2-star is a node with two arcs, etc.
- $T(y)$ — the number of triangles (3-clicks) in the graph

This model is very general, but not so good to work with: the N parameters are a large number, and the counts S_k are heavily dependent on each other, creating strong instability. We are also mostly interested in directed graphs (although the mapping between models for directed and indirected is usually simple).

As a practical approach inspired by this formulation, Wasserman & Patterson proposed the general Exponential Random Graph (ERG) model, also called p^*. Instead of specifying the statistics S_k, T above, they suggest a flexible framework where the user can define a set of statistics $u_1(y), \ldots, u_k(y)$ with corresponding parameters $\theta_1, \ldots, \theta_k$ and posit the model:

$$\mathbb{P}(Y = y) = \exp\left(\theta^T u(y) - \psi(\theta)\right),$$

where $\psi(\theta)$ is a normalization term (like the λ_{ij} above. The way to fit this model is with the same pseudo-likelihood approach, where modeling $\mathbb{P}(Y_{ij} = 1 | Y_{-ij})$ gives a simple logistic rergression in the parameters θ. The problems with this approach include the unreliability of PML and the strong dependence between summary statistics like number of k-stars. Some of the classical statistics that are included in $u(y)$:

1. **Edges**: The number of edges S_1
2. **Mutuality/reciprocity**: The number of directed pairs as in p_1
3. **Stransitivity**: the number of directed triangles in the graph

etc.

Latent space models

Now we switch to a different way of thinking about graphs. We assume the nodes have unobserved latent variables which are *locations* in some latent space, and they affect the affinity between nodes and their tendency to connect: closer nodes are more likely to connect. Formally, assume each node i has a latent (unobserved) location $Z_i \in \mathbb{R}^d$, and there is some distance (say Euclidean) on $\mathbb{R}^d$ such that $\mathbb{P}(Y_{ij} = 1)$ depends on $D(Z_i, Z, j)$.

We may also assumed that each edge has observed covariates X_{ij}, and nodes can have observed covariates too, in which case we assume things like $X_{ij} = X_i^T X_j$.

In this model we assume that given the latent variables the edges are independent, and the distribution has some parameters Θ:

$$\mathbb{P}\left(Y|Z,X;\Theta\right) = \prod_{i\neq j} \mathbb{P}(Y_{ij}|Z_i,Z_j,X_{ij};\Theta),$$

and typically assume simply a logistic model for the node probabilities:

$$\text{logit}\left(\mathbb{P}(Y_{ij}|Z_i,Z_j,X_{ij};\Theta)\right) = \alpha + \beta^T X_{ij} - D(Z_i,Z_j),$$

where $D(Z_i,Z_j) = \|Z_i - Z_j\|_2$ for example.

In the simple case that there are no observed features X_{ij} this takes the form:

$$\text{logit}\left(\mathbb{P}(Y_{ij}) = \alpha - D(Z_i,Z_j).\right.$$

Formally this ia missing data problem (since the Z_i are unobserved). Solving this maximum likelihood involves integrating over the unobserved variables, this is typically done by MCMC. It can lead to parameter estimates $\hat{\alpha}, \hat{\beta}$, but often we want to make use of these approaches for clustering or other ways to learn about structures in the latent space. For this we can infer the "likely" locations Z_i from the MCMC and apply clustering to them.

A more direct approach is to assume that there is a natural clustering model that generated the Z_i's and actually fit the parameters of this model as well as the parameters for $\mathbb{P}(y|Z)$ above. The simplest and most common approach is a Gaussian Mixture Model (GMM) assumption: $Z_i \sim \sum_k p_k N(\mu_k, \sigma_k^2 I)$, which assumes a collection of spherical Gaussians generated the latent locations. The number of Gaussians is the number of clusters, and we can think of this approach as combining the latent space modeling approach with the GMM approach for clustering. This is the approach of Handcock et al. (2002).

To figure out whether the model we found fits the data well, we have to consider both the GMM likelihood of the locations we inferred (which now also has parameters) and the likelihood of the observed data given the locations and parameters. We also have to penalize for the number of parameters as we always do. The theory behind the approximations that Handcock et al. (2002) employ is complex, but they come up with an approximate model selection measure based on BIC. We will not go into details, but accept that these are usable but not very reliable measures of how well the model fits and they can help us (together with visual and intuitive arguments) to find what models fit our data.

Scale-free networks

There is strong folklore that large "natural" networks (the internet, Facebook...) have typical properties:

1. Small number of nodes with a large or huge number of edges ("hubs")
2. Most nodes have very few edges (long tailed phenomenon)
3. The few-edges nodes are strongly clustered (communities)

It has been widely argued that for properties 1+2 a good fit is the scale-free model for the number of edges of each node, where

$$\mathbb{P}(X = k) \propto k^{-\gamma},$$

where $2 < \gamma \leq 3$: note that for $\gamma \leq 2$ there is no expectation, while for $\gamma \leq 3$ there is no variance. Hence we are assuming that these fat-tailed distributions in the scale-free network have an expectation but no variance.

Why is this called scale-free? Note that with this form:

$$\frac{\mathbb{P}(X = c \cdot k)}{\mathbb{P}(X = k)} = c^{-\gamma},$$

regardless of k, so the tail behavior is the same for small relative to medium, medium relative to large, etc.

Other "soft" properties of scale-free networks:

1. *Small world:* there are short paths from each node to each node going through hubs
2. *Robustness:* deleting nodes does not hurt connectivity or typical path lengthes
3. *Clustering:* formation of tight communities

Assuming we accept the fundamental importance of scale-free graphs (today widely disputed), we can ask what type of random processes can create such graphs? One important one is the **Preferential Attachment** model of Barabasi-Albert (1999). This simple model evolves as:

- Start with a set of m_0 nodes, randomly connected between them
- At stage N we add another node and connect it to $m < m_N$ existing nodes, with probability that is proportional to the number of connections each of them already has:

$$p_i \propto \frac{k_i}{\sum_j k_j}.$$

For this simple process they show:

- As the network grows we get $\mathbb{P}(k) \propto k^{-3}$, at the edge of the range for scale-free.
- The length of an average path is about $\frac{\log(N)}{\log\log(N)}$ when the network has N nodes $\Rightarrow$ a small world.

As mentioned above, in recent years there has been extensive skepticism about the usefulness of these models, and how well they fit real data.